Engineering Design 1 Technical Report
Group 5

Submitted to Professor Philip Jou
Stevens Institute of Technology
December 14, 2018

By: Ben Gilmore, Jose de la Cruz, Evan Eleftheriadis

I pledge my Honor that I have abided by the Stevens Honor System

Abstract

This technical report was written to explain the process that this group took to complete the final project for Engineering Design 1. This report will feature an introduction which will describe the goals and objectives of the course as well as the mission of the robot for the final test. The overall assembly will then be discussed, including the wiring, mechanical, and electrical design of the robot and why the group made certain decisions pertaining to the assembly of the robot. The two road tests as well as the final tests will be described as well as the challenges that were faced by the group. The conclusions of the project as well as certain attachments will be included at the end of the report.

Table of Contents

Introduction

Purpose:

The purpose of this technical report is to go over the entire process that the group went through to understand the robot, to assemble the robot, and to complete the final course. It will explain what was needed for each road test and the difficulties that were faced by the group to complete the challenges. The design of the robot will be covered, including the system design, mechanical design, software design, and electrical design. The report will cover the testing of the systems and the code and the process by which the group completed the project.

Background and Objectives:

The purpose of the project is to have the group work together as an efficient team under a specific deadline acquiring engineering skills to be used to become familiar with the robot and all of its components. The team began by getting familiar with the equipment and tools in the lab. Once the robot kit was received the group studied the robot to learn the purpose of the individual parts and to understand how each component worked. The team then assembled the robot while taking the various quizzes associated with it.

Summary of Project Specifications:

The team was only allowed to use materials that were given to them by the instructor. The group is not allowed to add any additional modifications to it. The robot is not allowed to exceed six inches in height. The first road test that needed to be completed required the robot to go in a straight line and stop within two inches of the opposite wall(see Figure 1-1). The group was able to get this done after a few minor setbacks. The next road test was the left hand loop, which required the robot to complete three laps around the center wall in the track in as little time as possible(see Figure 1-2).

Mission of Robot:

After the group completed the two road tests work began on the final test. The final tests requires the robot to start in one of four starting positions to be known on the day of the test and then after a ten to fifteen second delay proceed through the course and hit the bumpers in the order of BDAC(see Figure 1-3). After the last bumper is hit the robot will return to its starting location. The time allotted for the final test is 120 seconds.

Approach to Project:

The first thing that the group needed to do was figure out what each member would do to complete the project. It was decided that Jose would be the lead coder while Evan and Ben would be the ones to assemble the robot. Before assembling the group needed to understand what every component did, which was done by taking the quizzes. The group then needed to come up with a plan on how to code the robot for the road tests. This was done through a series of trial and error, changing up the code a little bit at a time until it was just right. Once the road tests were completed the group needed to focus on the final test. It was decided to not use the sensors for the final as it was easier to code the path that the robot would take.

Roles of Each Group Member:

All group members worked on the assembly of the robot. After the initial assembly Evan and Ben took over to attach all of the sensors and the boards for the robot. It was also the responsibility of Evan and Ben to complete the quizzes, including the measurements, programming, and assembly quiz. Jose was the lead coder of the group and was responsibility for calibrating the motors as well as the sensors. Ben helped Jose with the code by helping with the small adjustments that were needed during the final test.

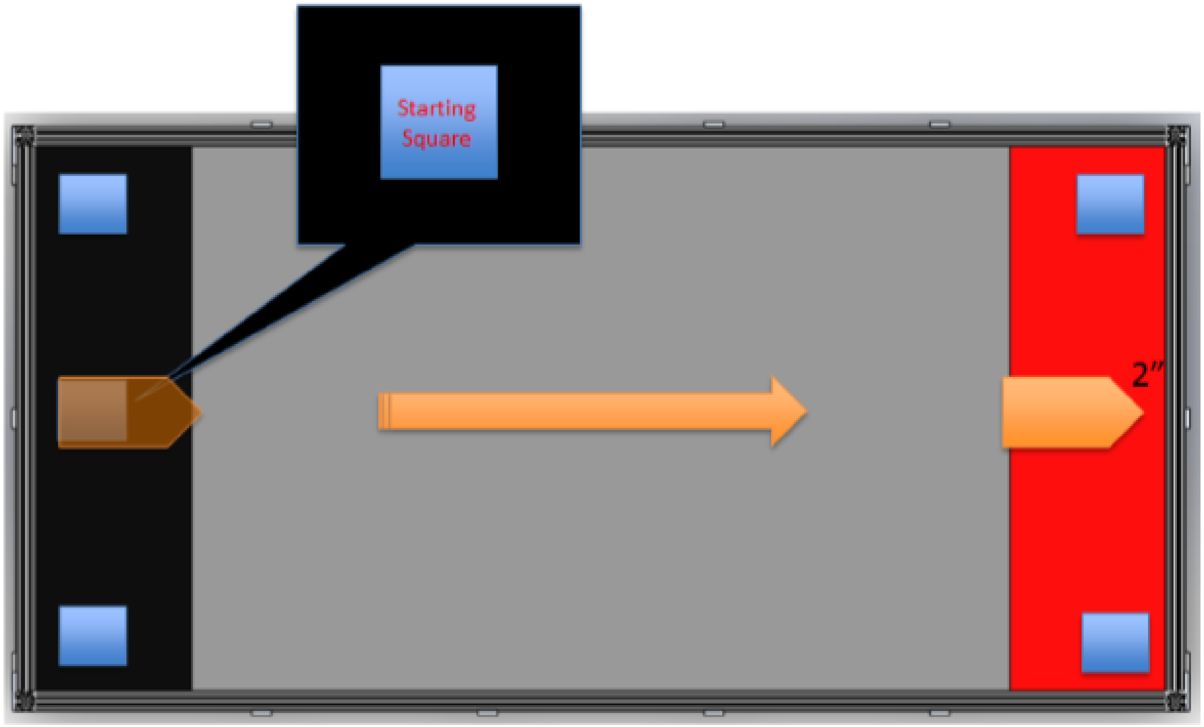


Figure 1-1: The first road test, pictured above, required the robot to move in a straight line.

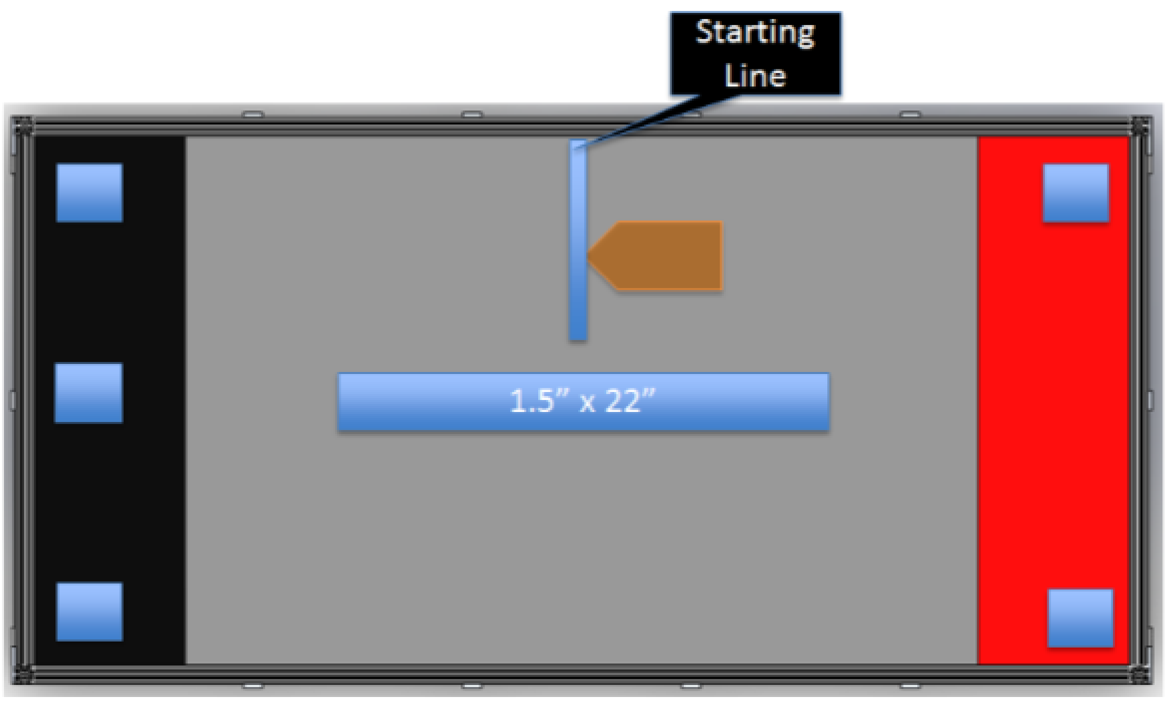


Figure 1-2: The second road test, pictured above, required the robot to complete three left hand loops around the central wall.

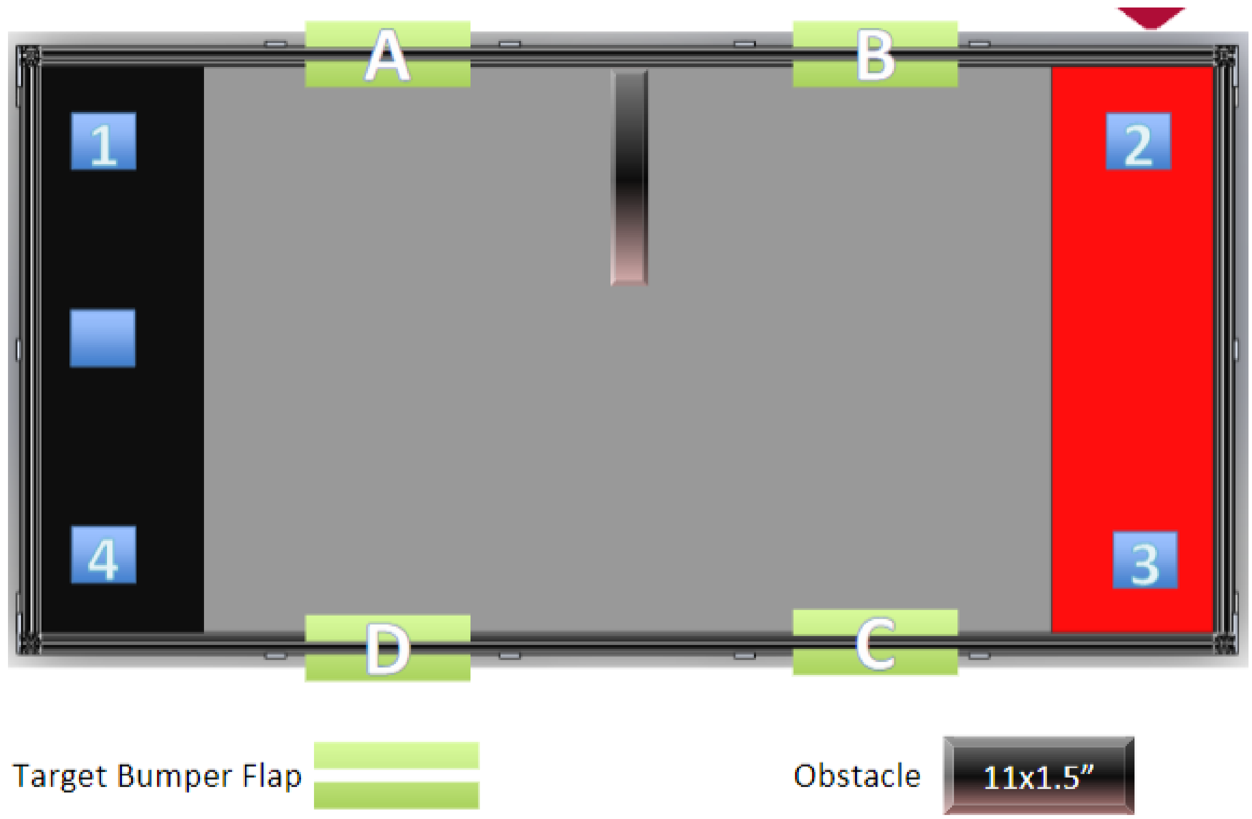


Figure 1-3: The final test will take place on the arena pictured above. The robot will be required to start on any of the four starting locations, wait for 10-15 seconds and then complete the course hitting the bumpers in the order BDAC. The robot will then return to its starting location after hitting the final bumper.

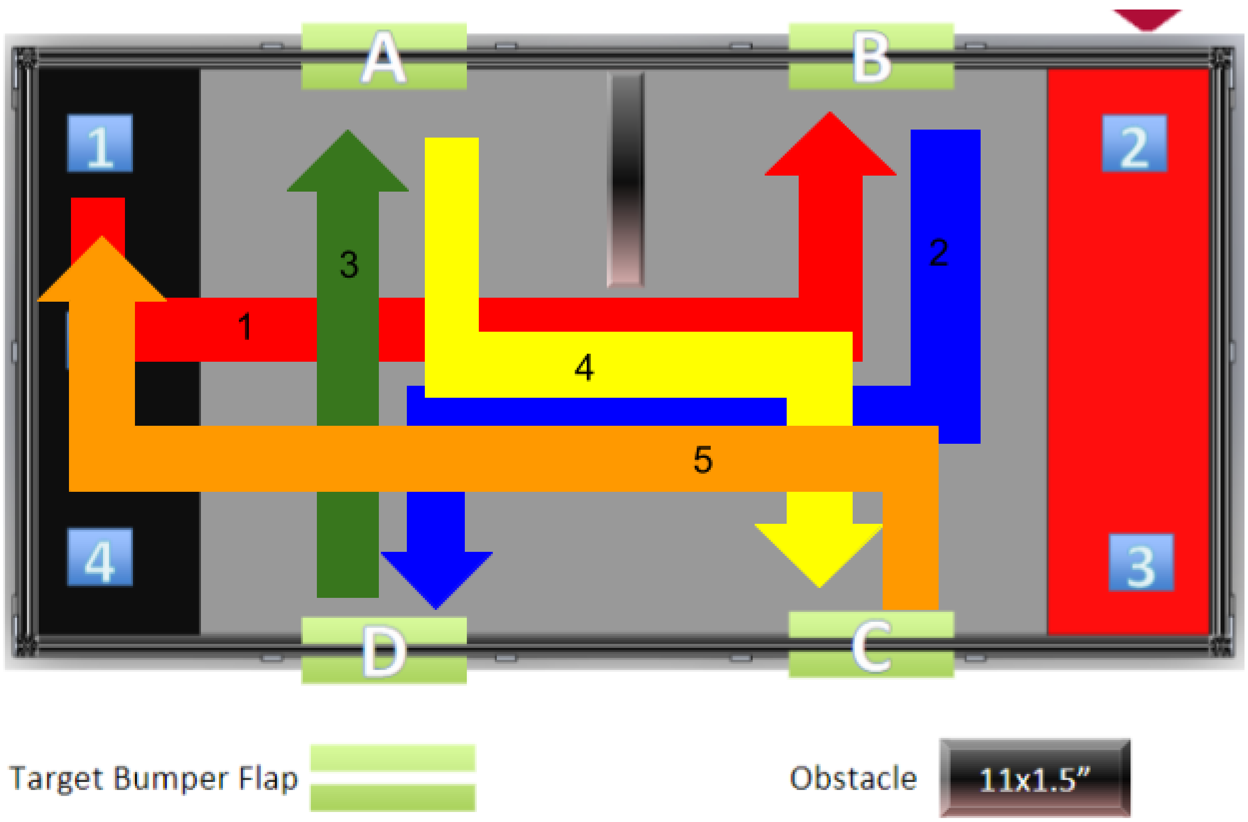
Discussion

Requirements:

The requirements of the final test was given to the group by the professor. The robot will be able to start in any of the four starting locations, as it will not be given until the day of the final test. Once the robot is placed down, it should have a delay of about ten to fifteen seconds before it begins to move through the arena. Each group was given a specific order that the robot has to hit the bumpers in. This groups order is BDAC. This means that the first bumper that the robot must hit is the B bumper. It will then progress through the rest of the bumpers in the correct order at a relatively quick pace. Once the robot hits the last bumper it will then return to the original starting position. As the group did the first two road tests, it became clear that the design of the robot needed to change to better suit the requirements of the final test. The robot had only one sensor for the road test as it only needed to move forward and stop once it got into the correct position. When the group moved onto the second test the sensor was moved to the side so that the robot could complete the loop. It was then made clear that additional sensors were needed so that the robot could accomplish the task of moving forward and turning. This required the addition of two more sensors for a total of three, one of the front and two on either side of the robot.

Figure 2-1:

If the starting location was set to be 1 then the robot will move forward until it got past the wall in the center and then turn to the left and move forward until it got to where the B bumper is. From there it turns left again and move forward until it hits the B bumper, shown by a red arrow. It will then reverse back the way it the came and reverse right so that it is now facing in the direction of the D bumper. It moves forward and then turns left and move forward to hit the D bumper, shown by a blue arrow. The robot then reverses to hit the A bumper, shown by a green arrow, and then reverses and turns to the right to move toward the C bumper where it turns right and hits the C bumper, shown by a yellow arrow. The robot then goes back the way it came to return to the initial position, shown by an orange arrow.



System Design:

There are two sensors on the bracket towards the back of the robot which were used in the left hand loop test, located right above the board. The other sensor is located at the front of the robot. This will be used to judge the distance in front of the robot. It is on the breadboard that is attached to the arduino board.

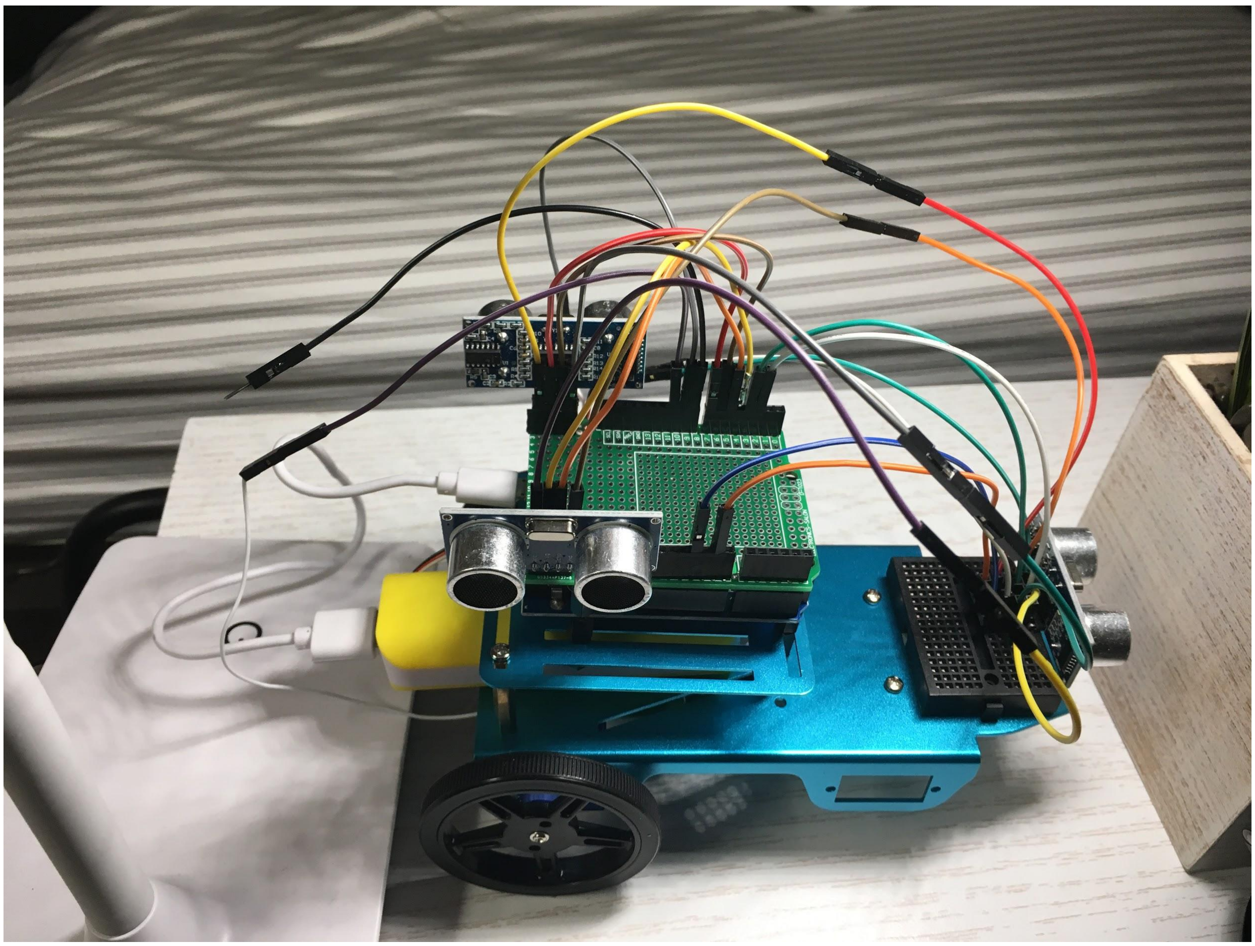


Figure 3.1: The structure of the robot and the positioning of the sensors can be seen above

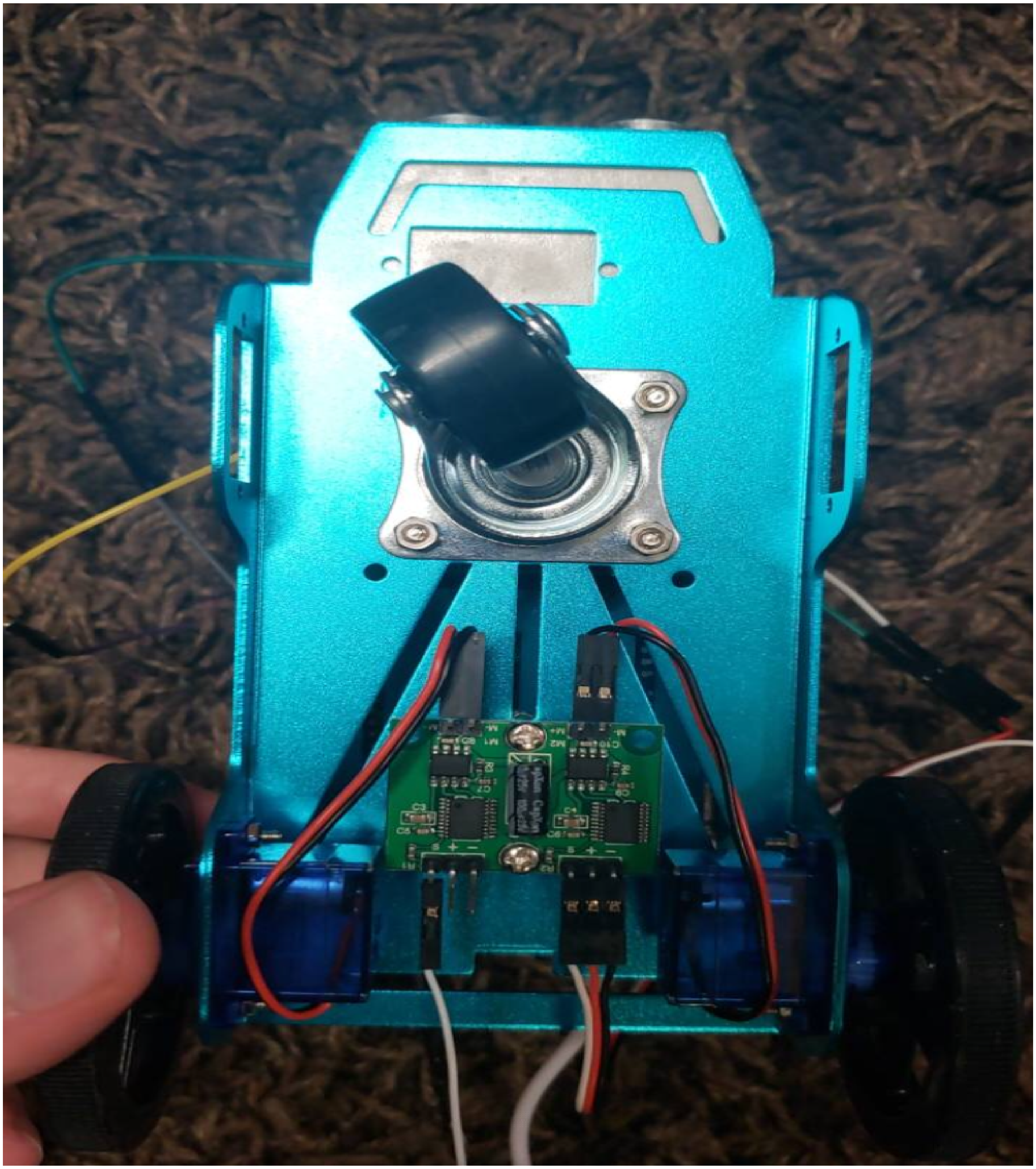


Figure 3.2: The positioning of the motors as well as the free spinning center wheel.

Software Mechanics:

The robot navigates using three sensors on its chassis. The sensor works like echolocation or radar. The trigger pin sends a signal outwards, while the echo pin relays the signal as it bounces back to the robot. Data is returned as a integer that can be used to measure distance. Using the integer relayed back to the arduino board, the data is converted to the motors which in place allow the robot to proceed in the wanted direction/ acceleration.

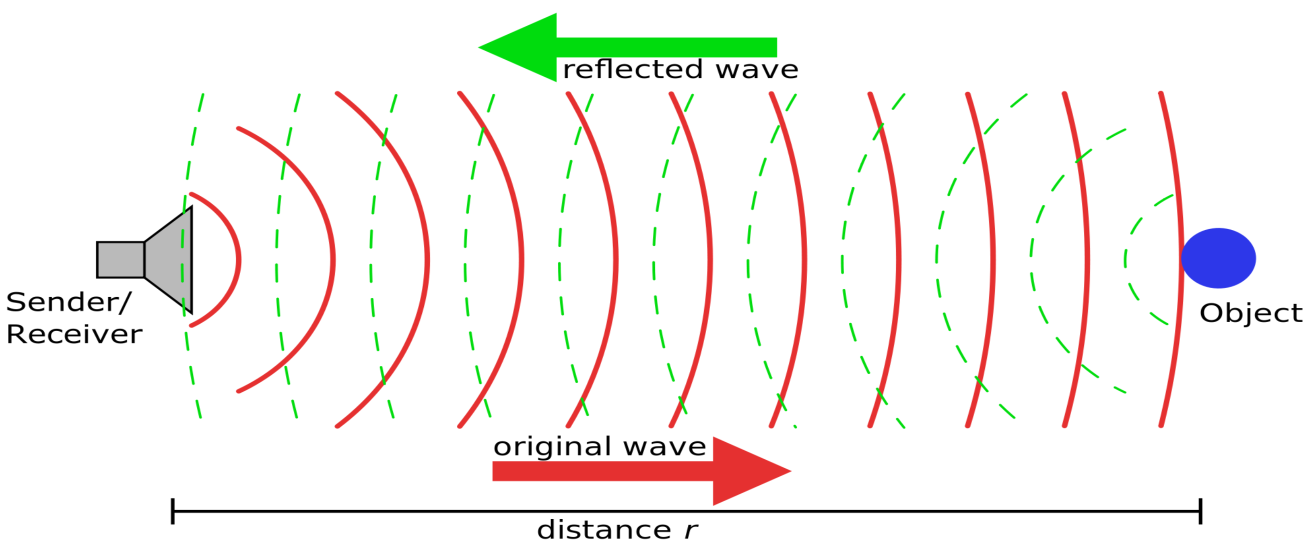


Figure 4.1: The diagram above illustrates the way the sensors work. Waves are transmitted to an object and then bounce off and relay the information back to the sensor's receiver.

Mechanical Design:

Due to the fact the motors are placed in opposite directions, one code must have the first wheel spinning forwards and the other spinning in reverse. For the coding, force of friction and error within the motors must be taken into account when inputting the values into the code for the motor's acceleration/ deceleration.

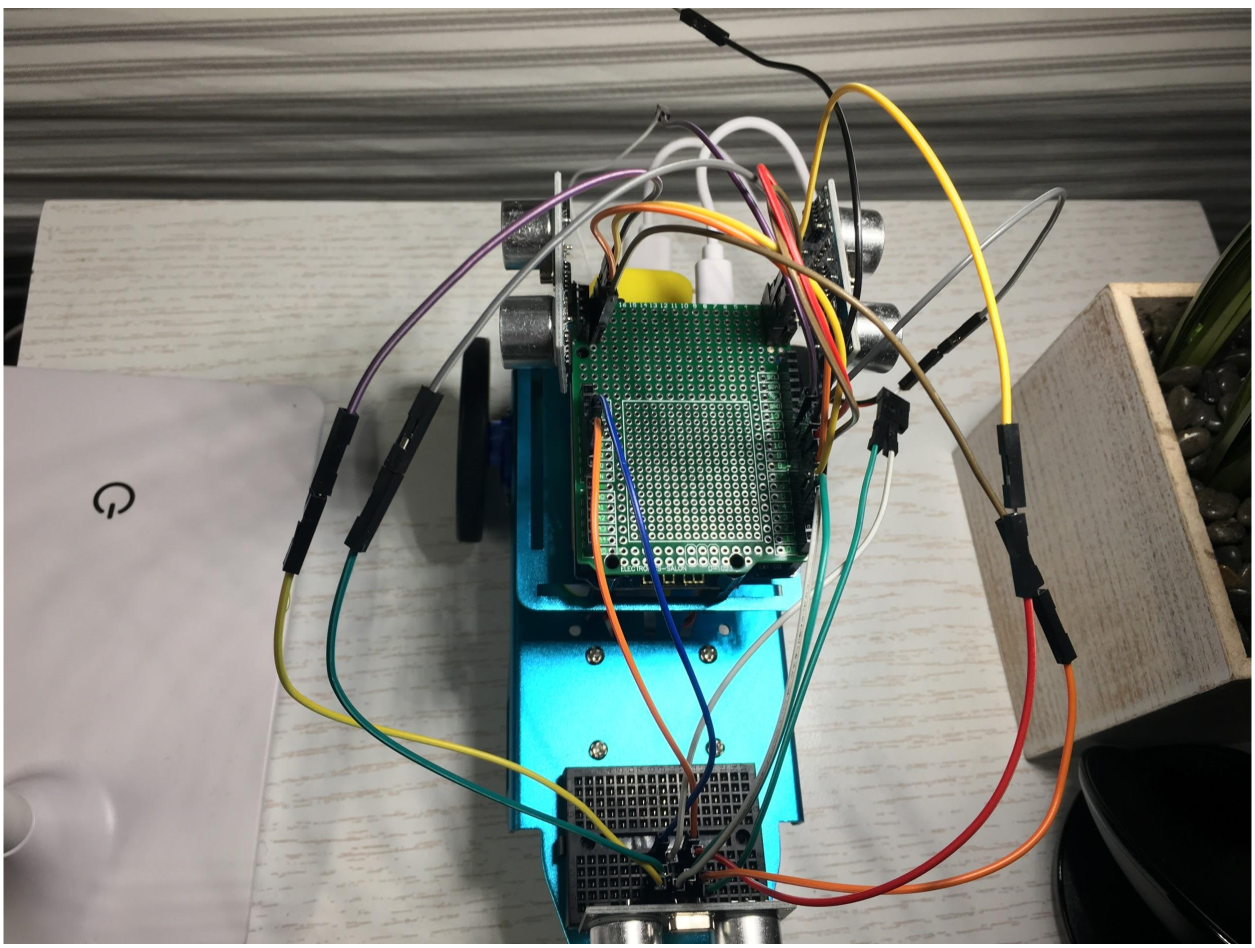


Figure 5.1: the positioning of wires relaying information from sensors to arduino board to the motors.

Software Design and Coding:

All of the coding involved in this project was done within the Arduino IDE. Our robot used a WeMos board that work within the Arduino Architecture after installing both the specific hardware package for the Arduino IDE to be able to upload to the WeMos Board and the WeMos specific diver that would allow the computer with the Arduino code to connect to the Wemos board. Once this was done the Arduino IDE had to be told which specific board to use, the WeMos D1R1 for the team. Finally, we needed to sync our serial rate with the WeMos board. Once this was done the Wemos Board could finally be used with our robot.

In order for the WeMos board to interface with the motor controller the servo library built into the Arduino IDE was included in the code. An object needs to be created for each motor we used so we made two objects called left_motor and right_motor which refer to the motors to the left and right of our robot when looking at it from the back. The Servos must be assigned to a pin on the WeMos Board but once this is done the servos are operationable. Controlling the Motors is very un-intuitive because of the odd way the code needs to be changed to make the motors go in reverse.

Function	(Parameter)	Result
write	0	Full speed
	90	Stop
	180	Full speed opposite direction

This system lead to much confusion even when just trying to make the robot go straight especially because the motors end up going opposite directions when the same number is inputted to both. To combat this the left motor was wired in reverse so that 180 meant forward for both motors. Getting the motors to go straight was a struggle because of extreme inconsistencies between the motors. In the final version of the code used for the last road test getting the robot to go straight involved having one motor going at full speed and another nearly stopped.

Using the UltraSonic Sensors was a lot easier than the motors. Each sensor takes up two pins one output and one input. Once these are assigned the function ultrasonicPing(Output,Input) can be used to obtain the amount of microseconds it took for the ultrasonic pulse to get back to the sensor. In the end, 3 ultrasonic sensors were used and the values were stored in 3 variables.

```
pinMode(D2,OUTPUT);// trib Front
pinMode(D3,INPUT);// echo Front
pinMode(D4,OUTPUT); //trig Right
pinMode(D5,INPUT); //echo Right
pinMode(D6,OUTPUT); //trig Left
pinMode(D7,INPUT); //echo Left

long front = ultrasonicPing(D2,D3); //distance between sensor and wall
long right = ultrasonicPing(D4,D5); //distance between sensor and wall
long left = ultrasonicPing(D6,D7); //distance between sensor and wall
```

```
void go(int x) // Robot goes Straight
{
  left_motor.write(73);
  right_motor.write(0);
  delay(x);
  left_motor.write(90);
  right_motor.write(90);
  delay(1000);
}
void reverse(int x) // Robot Reverses
{
  left_motor.write(110);
  right_motor.write(160);
  delay(x);
  left_motor.write(90);
  right_motor.write(90);
  delay(1000);
}
void less_left() // Robot Turns left 90 degrees
{
  left_motor.write(90);
  right_motor.write(83);
  delay(1450);
  left_motor.write(90);
  right_motor.write(90);
  delay(1000);
}
void turn_right() // Robot Turns right 90 degrees
{
  left_motor.write(84);
  right_motor.write(90);
  delay(1400);
  left_motor.write(90);
  right_motor.write(90);
  delay(1000);
```

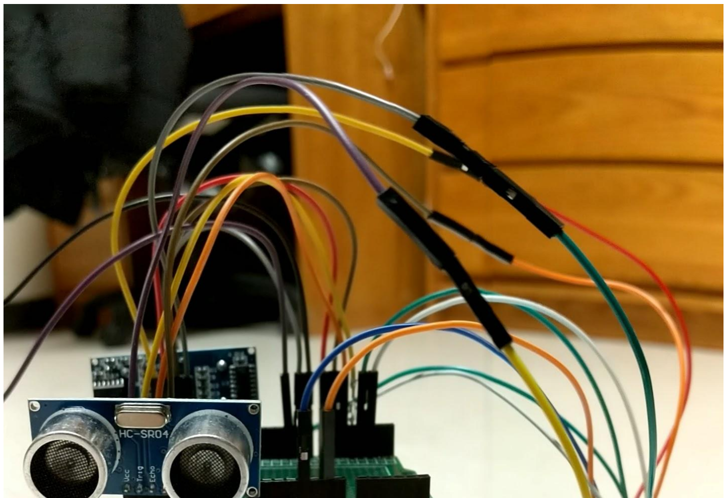
By the time we made the code for road test 3 we were using functions to make the code much more readable and adaptable. As the days went on in class it was clear that sometimes the battery would warm up and start to perform better or begin to die and perform worse. The easiest way to combat this was to edit individual functions that corresponded to the robots actions. When the battery was over performing the values on the functions needed to be toned down a little and vice versa when the battery began dying.

The go and reverse functions took an integer as an argument which would tell the robot how long to go forward or reverse while the others were functions that could simply be called to have the robot commit the action.

In the end all of these functions were put into one large function that the Wemos board could run in order to complete Road Test 3.

```
void from_3()
{
  delay(800);
  go(600); //from spot to middle
  less_left();
  go(700); // from spot to pad
  turn_right();
  go(1100); // from middle to pad
  // B
  reverse(1000); // REVERSE from pad to middle WIDTH
  less_left();
  go(1300); // from pad to pad LENGTH
  less_left(); //LESS LEFT
  go(1500); // from middle to pad WIDTH
  // D
  reverse(2000); // REVERSE from pad to pad WIDTH
  // A
  go(800); //from pad to middle
  less_left();
  go(1600); //from pad to pad LENTH
  turn_right();
  go(1000); //from middle to pad WIDTH
  //C
  reverse(400);
  less_left();
  go(650);
  turn_right();
  go(400);//from pad to spot
}
```

Electrical and Wiring Design:



The electrical and wiring design is a place our robot is lacking. The wires are needlessly long and tend to get in the way of handling the robot and even get in the way sometimes when the robot moves around. This problem was partially alleviated by the board that we put on top of the WeMos board. This allowed two of our sensors to be put in a place that a bread board wouldn't allow. The wires that came from the motors posed the biggest problem because of how often they ran into the wheels.

The pin configuration of our robot was pretty simple. Sensors went first in a Output then Input order. Second was our motors which needed a pin each. Left motor went first followed closely by right motor in pin order.

Pin	Function
2	Front Sensor Output
3	Front Sensor Input
4	Right Sensor Output
5	Right Sensor Input
6	Left Sensor Output
7	Left Sensor Input
9	Left Motor
10	Right Motor

Integration Testing and Evaluation:

The group was required to complete two road tests before they were able to move onto the final test. These two tests were instrumental in the testing of the design of the robot and how well each part performed. The group quickly realized that the two motors were not as powerful as each other, so the numbers to set the speed required variation to make sure the robot went straight. This required multiple trials of having the robot go straight and making small adjustments to the speed of each motor after each one. Eventually the group was able to get both motors to work in tandem. Unfortunately this had to be done twice as the robot was lost and the group was required to use a fresh robot. This meant that the team needed to calibrate the motors again. Although luckily the group had experience so the second time was much quicker. These tests were also used to see how well the sensors worked and to determine where the group should mount them on the robot for peak efficiency. For the first test only one sensor was used. The sensor was then moved to the side to complete the loop. The final design of the robot uses three sensors but these were not used in the final test. The sensors were not as reliable as wanted so the group decided to hardcode the path of the robot so that it hit the bumpers in the required order. The group did not take the sensors off the robot after it was decided that they would not be needed because there was no reason to take them off. Also, the weight of the robot was centered using the sensors so that it continued along a straight path as well as being an aesthetic choice of the team.

Final Demonstration:

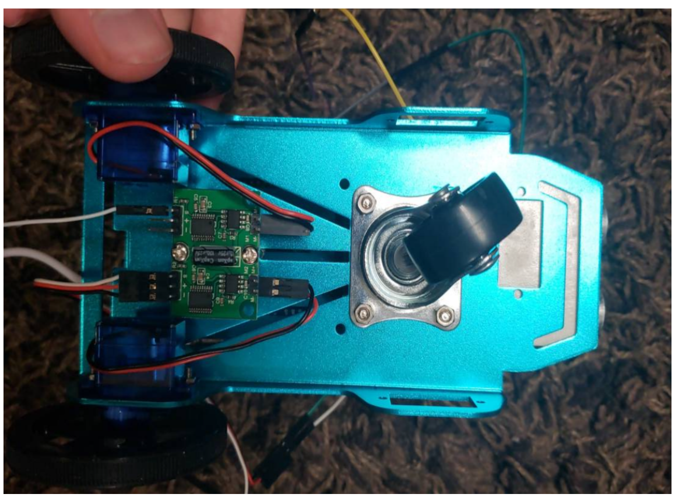
The final test of the robot was a success. The group took almost the entire time of the last class to perfect the code so that the robot performed the way it was supposed to. The robot began in starting position 3. From there it went forward and tuned left then right to hit the first bumper. It then reversed and turned to the right and made its way to the other side of the arena. The robot turned to the left and moved forward until it hit the second bumper and then reversed to hit the third bumper. It then went forward, to the left, forward again, and to the right to hit the last bumper. It then turned left and moved forward to get back to initial starting position.

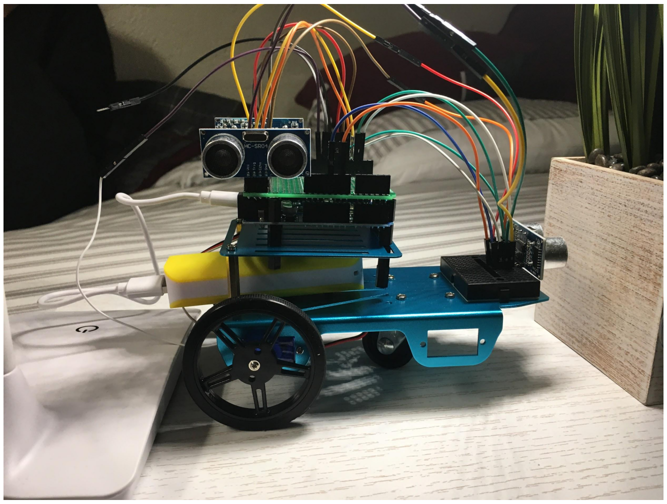
The group decided to write the code as the night progressed. So once a target was hit the code was added to move onto the next one. As stated before, the sensors were not used to complete the course. Instead the course was hard coded into the robot with the correct distances and speeds in order to reach each bumper. The four milestones were the four bumpers. The final run included the robot hitting all four bumpers and then returning to the starting position. The full run took about 45 seconds. The group finished the night with completing the course in its entirety as well as getting the style points.

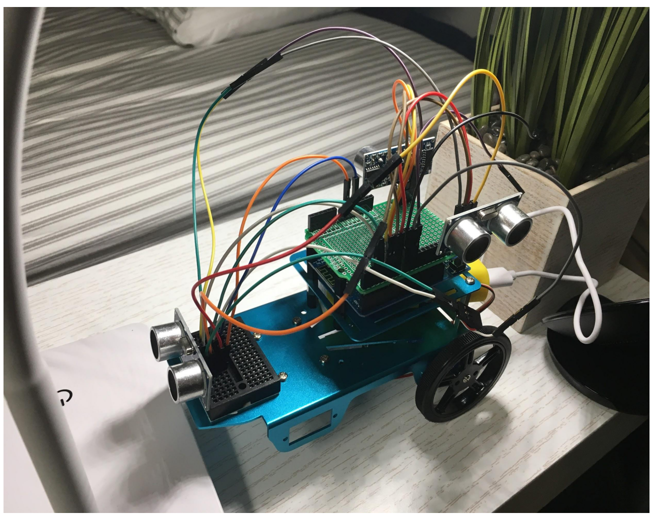
Conclusion

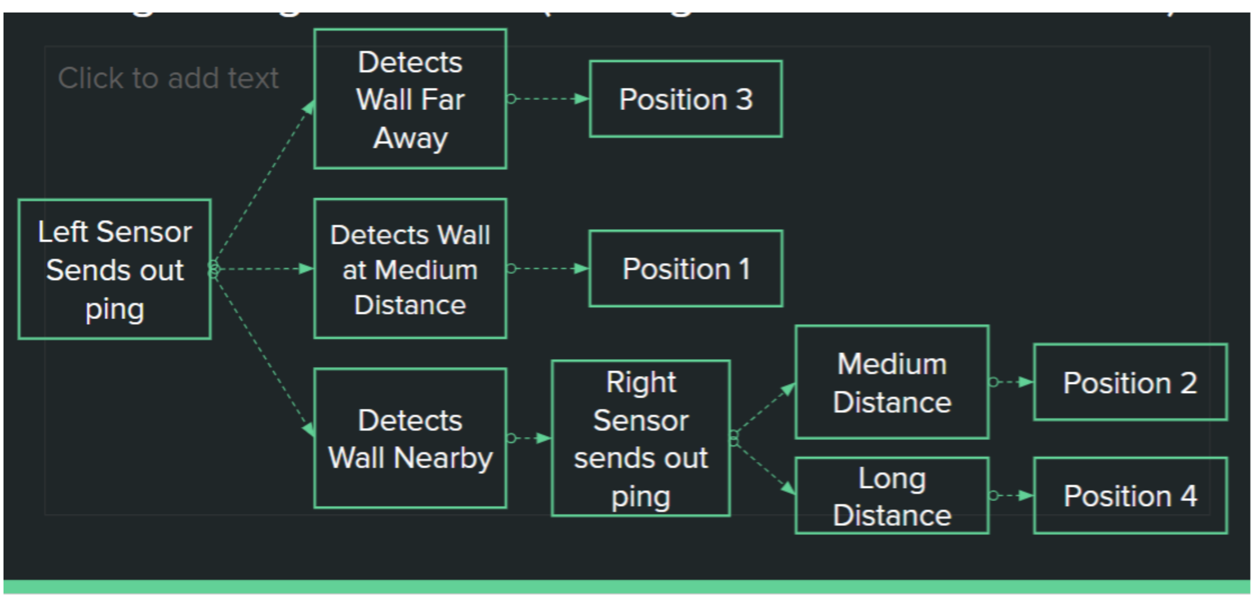
One of the sensors provided in the robot kit was faulty and didn't perform the expected action which caused the group to dedicate time in finding the error in the code which was correct from the start. The sensor would stop in time before a cardboard wall yet didn't recognize the glass surface in the nascar loop and continuously crashed into the wall. The wheels also may lay at a angle and cause unwanted degree changes in the robot's direction. The wheel without the motor in the front had to be aligned straight as to not throw the robot off its course. The robot was additionally lost in which it made the team lose a week in fine tuning the assembly but allowed the group to work on the code and the technical report

Attachments








Click to add text
Left Sensor Sends out ping
Detects Wall Far Away
Position 3
Detects Wall at Medium Distance
Position 1
Detects Wall Nearby
Right Sensor sends out ping
Medium Distance
Position 2
Long Distance
Position 4